Semantic and Instance Segmentation

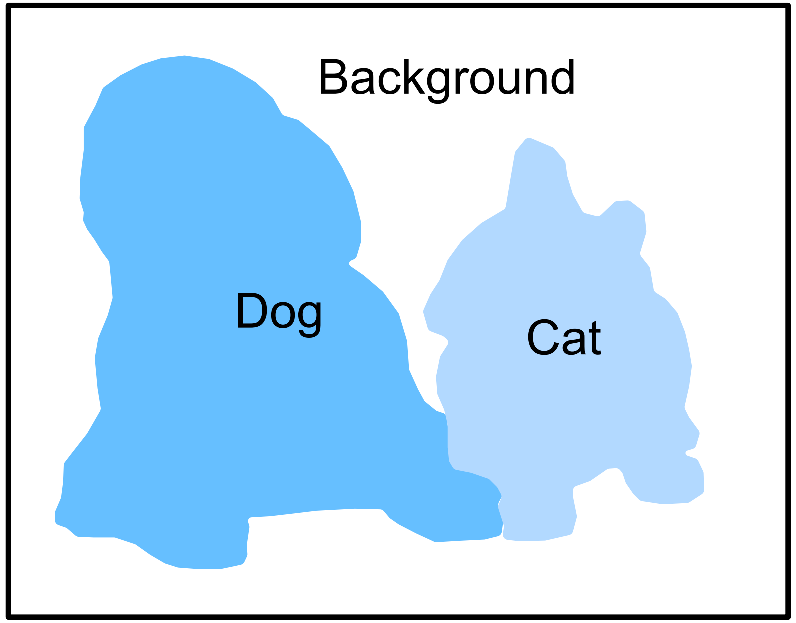
CMPUT 328

Nilanjan Ray

Semantic and Instance Segmentation

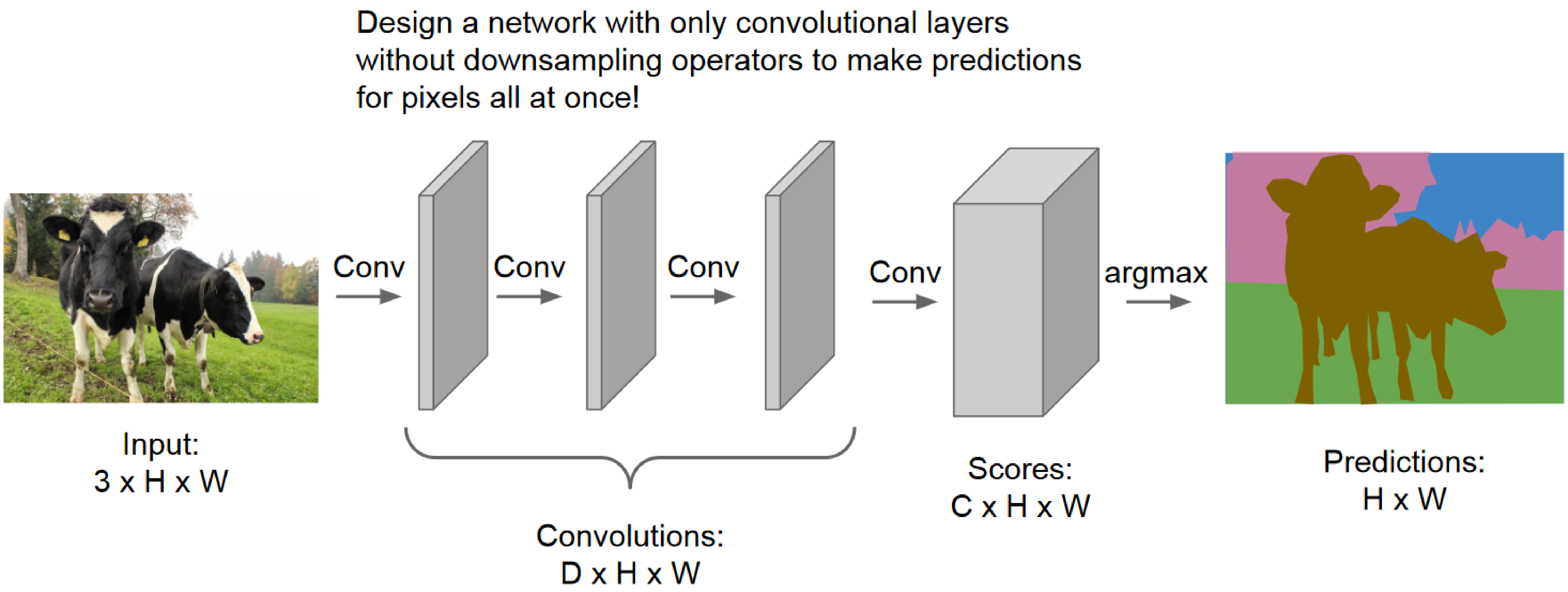
- Semantic segmentation is classifying each pixel of a picture into a category or class.
- Instance segmentation = Semantic segmentation + Object detection
- Let's explore these two tasks of computer vision in this lecture

Semantic segmentation



https://d2l.ai/chapter_computer-vision/semantic-segmentation-and-dataset.html

Use a fully convolutional architecture



Source: cs231n course

How would you train a fully convolutional net?

- Cross entropy loss at every pixel location and sum over all the pixel locations
- Let's understand how a training dataset is prepared for semantic segmentation

Input

segmented

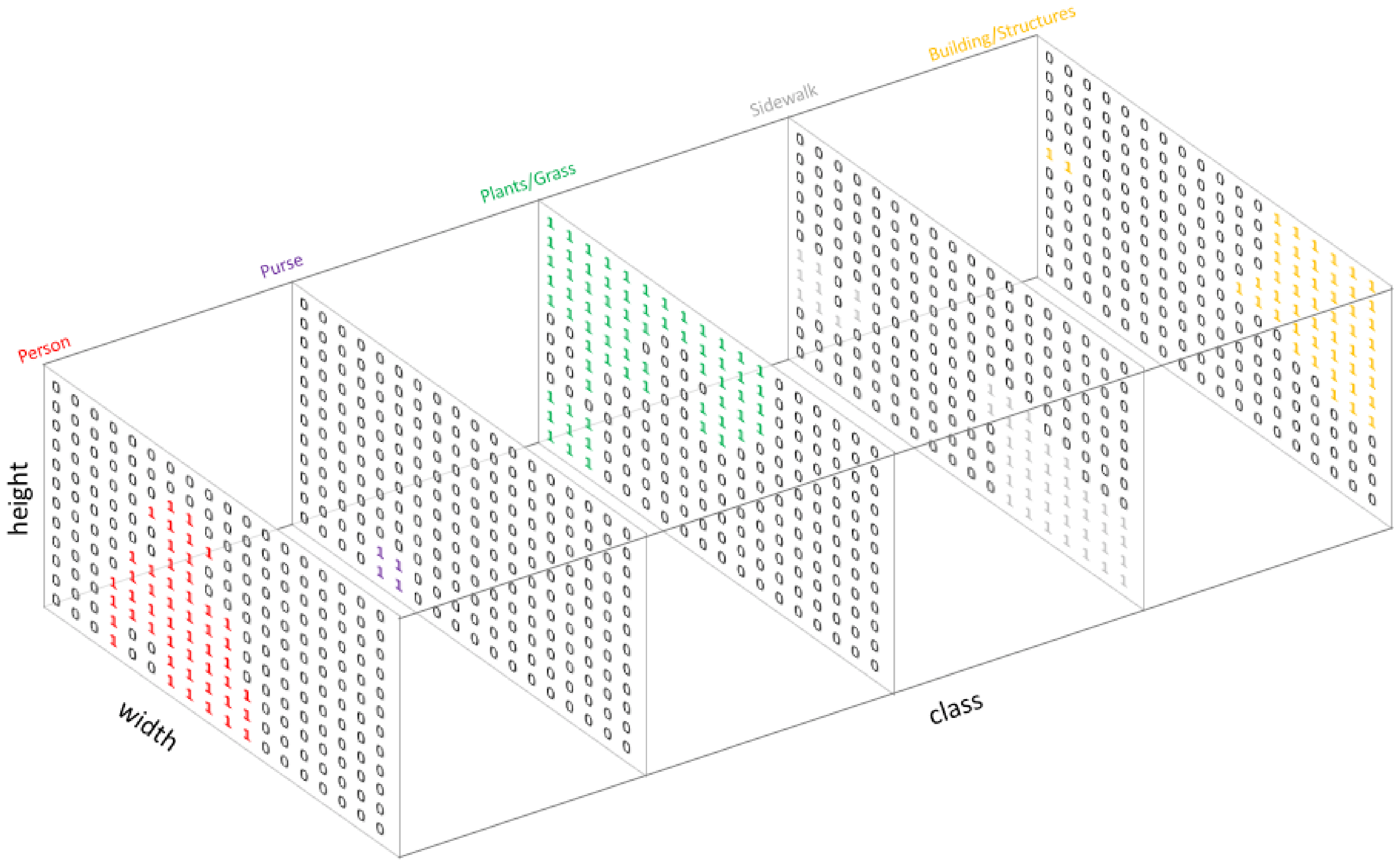
1: Person
2: Purse
3: Plants/Grass
4: Sidewalk
5: Building/Structures

3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	1	1	1	1	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	5	5	5	5	5	5	5
5	5	3	3	3	3	1	1	3	3	5	5	5	5	5	5	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	5	5	5	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	4	4	5	5	5
4	4	4	1	1	1	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	1	1	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	1	4	4	4	4	4	4	4

Semantic Labels

Source: https://www.jeremyjordan.me/semantic-segmentation/

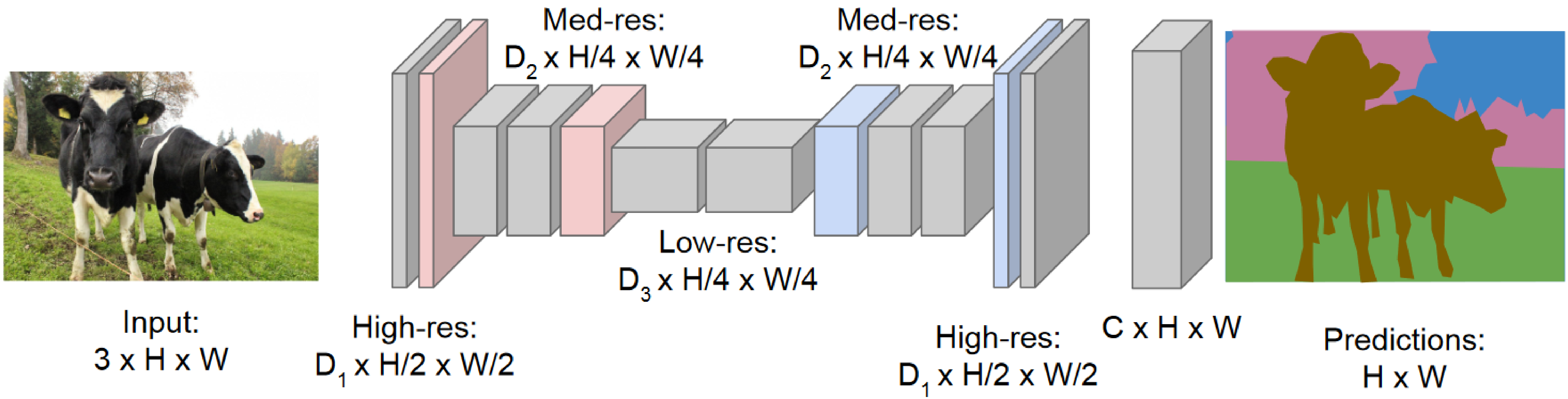
Labels for semantic segmentation



Source: https://www.jeremyjordan.me/semantic-segmentation/

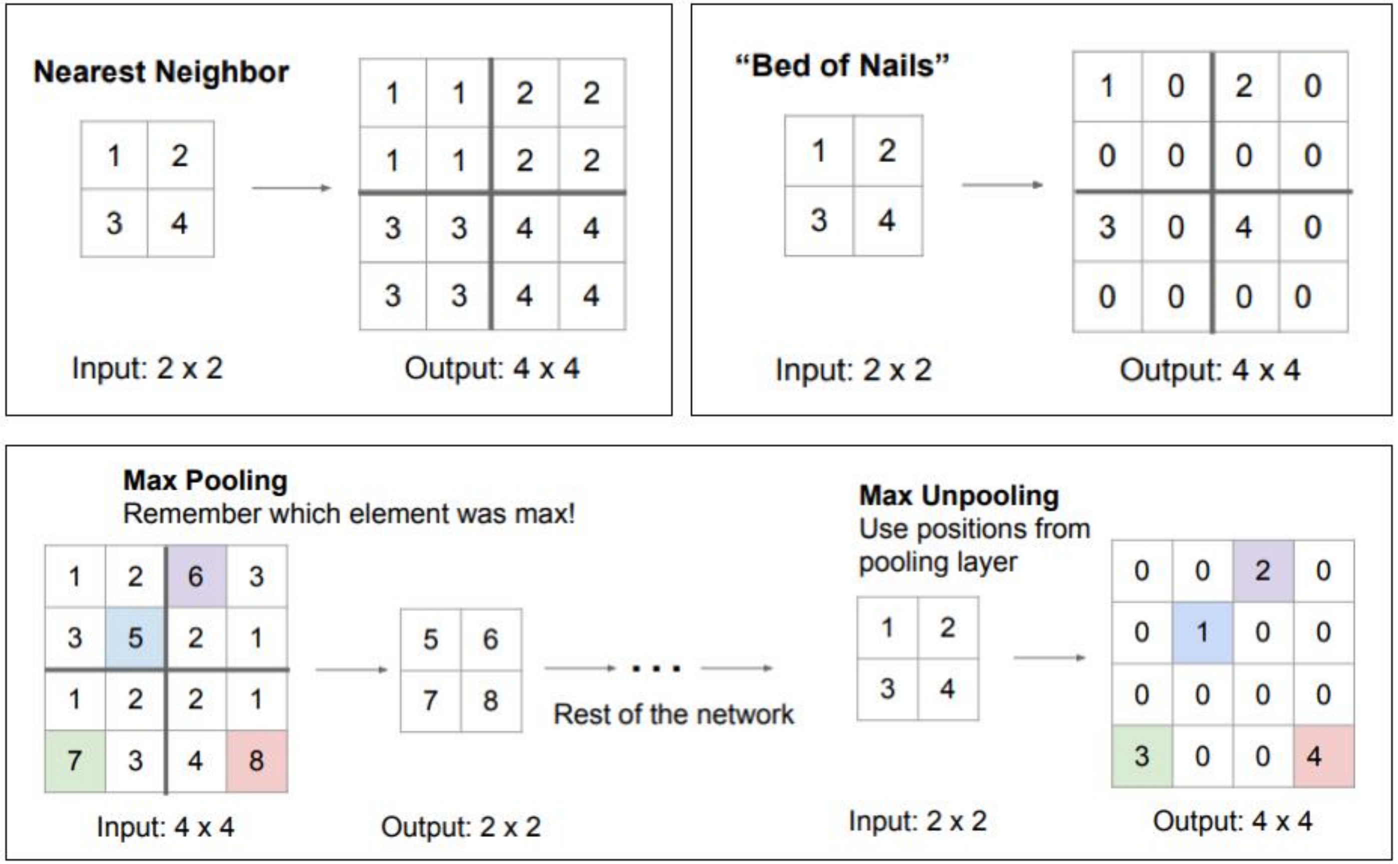
Downsampling in network architecture

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!



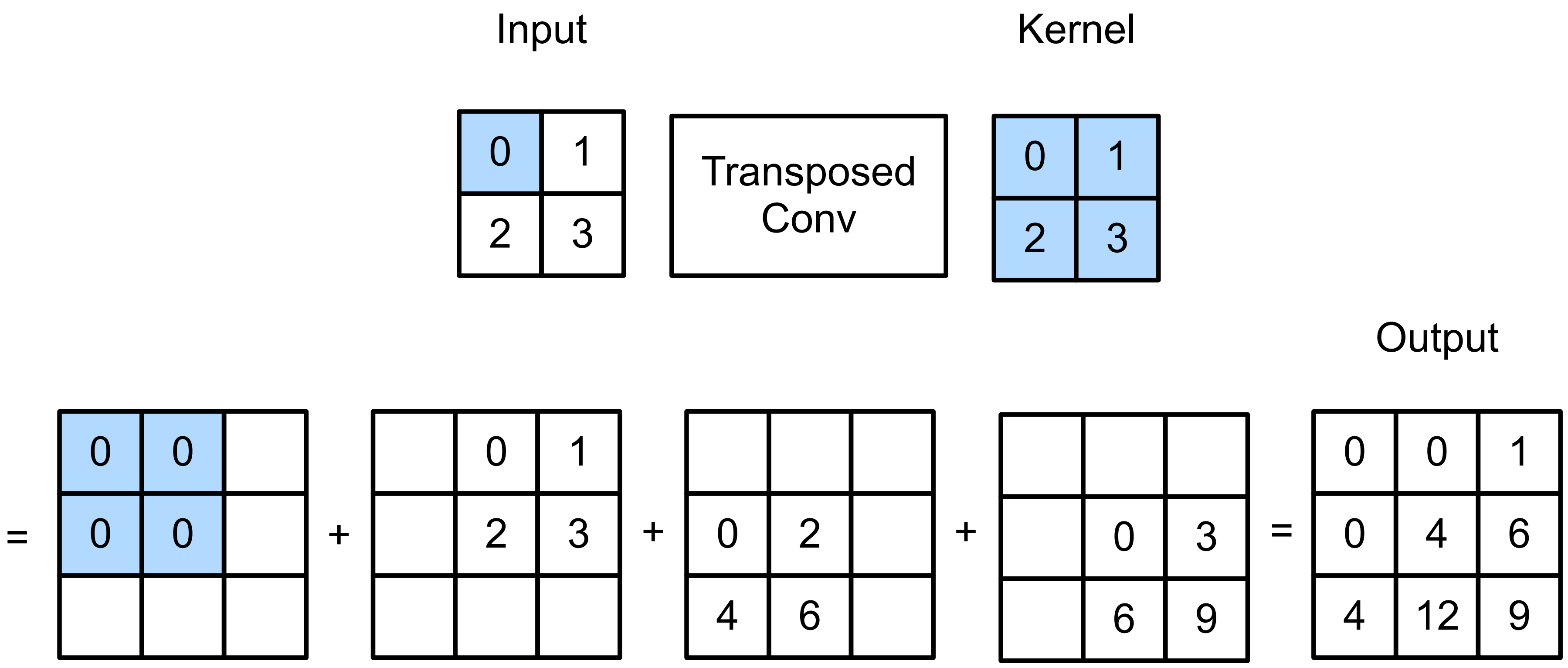
Downsampling leads to computational efficiency. Why does it not affect segmentation accuracy?

Upsampling: A few ideas



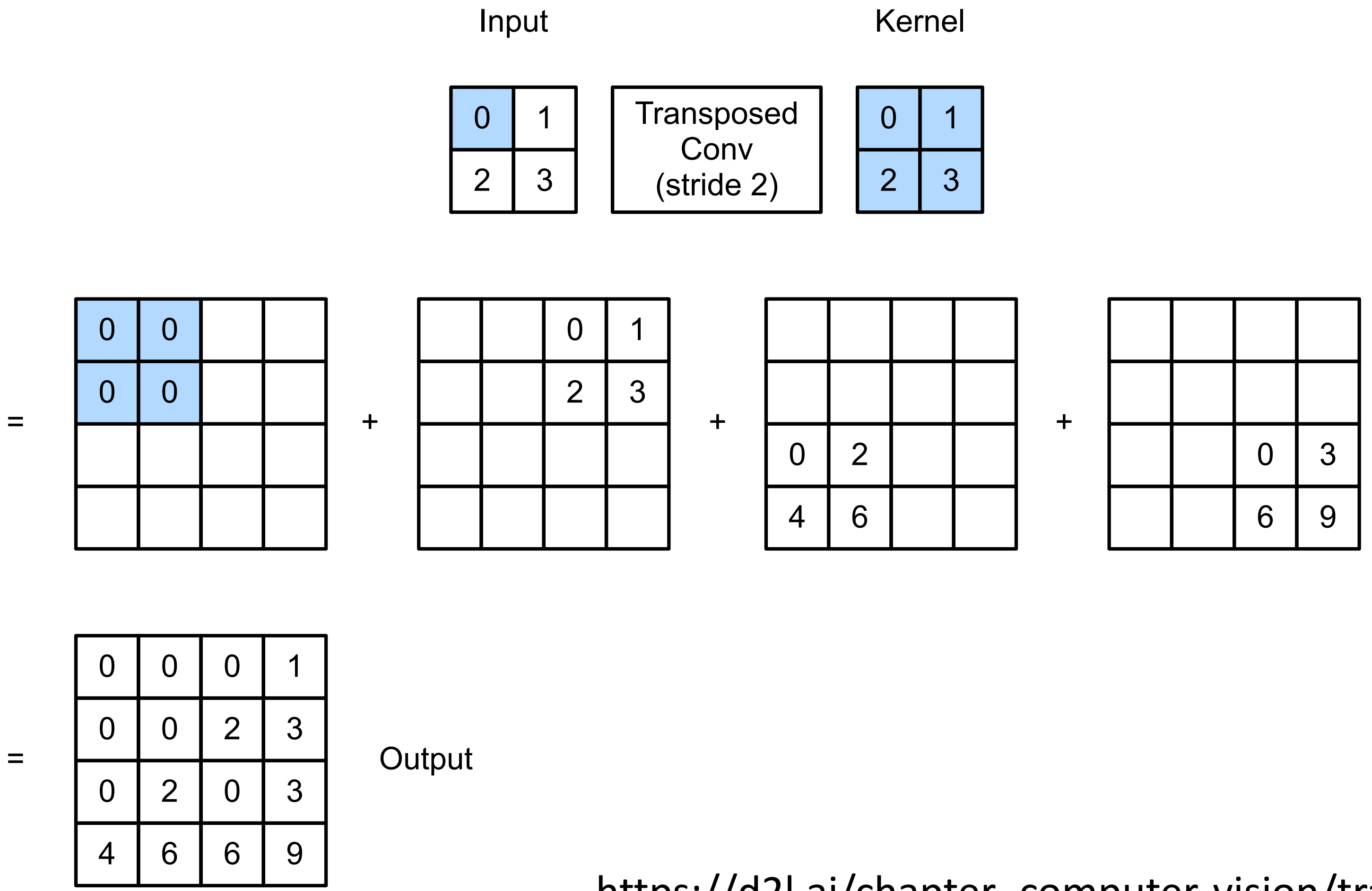
Source: https://www.jeremyjordan.me/semantic-segmentation/

Upsampling: Transposed convolution



Source:https://d2l.ai/chapter_computer-vision/transposed-conv.html

Transposed convolution: Another example



https://d2l.ai/chapter_computer-vision/transposed-conv.html

BTW, why is it called transposed convolution?

- An explanation is here: https://d2l.ai/chapter_computer-vision/transposed-conv.html

Upsampling: Yet another idea

- Normal convolution followed by bilinear interpolation

- Our experience is that it works better than transposed convolution, which sometimes produces a checkerboard type artifact.

An example fully convolutional architecture and its training

- https://d2l.ai/chapter_computer-vision/fcn.html

UNet: An important architecture for semantic segmentation

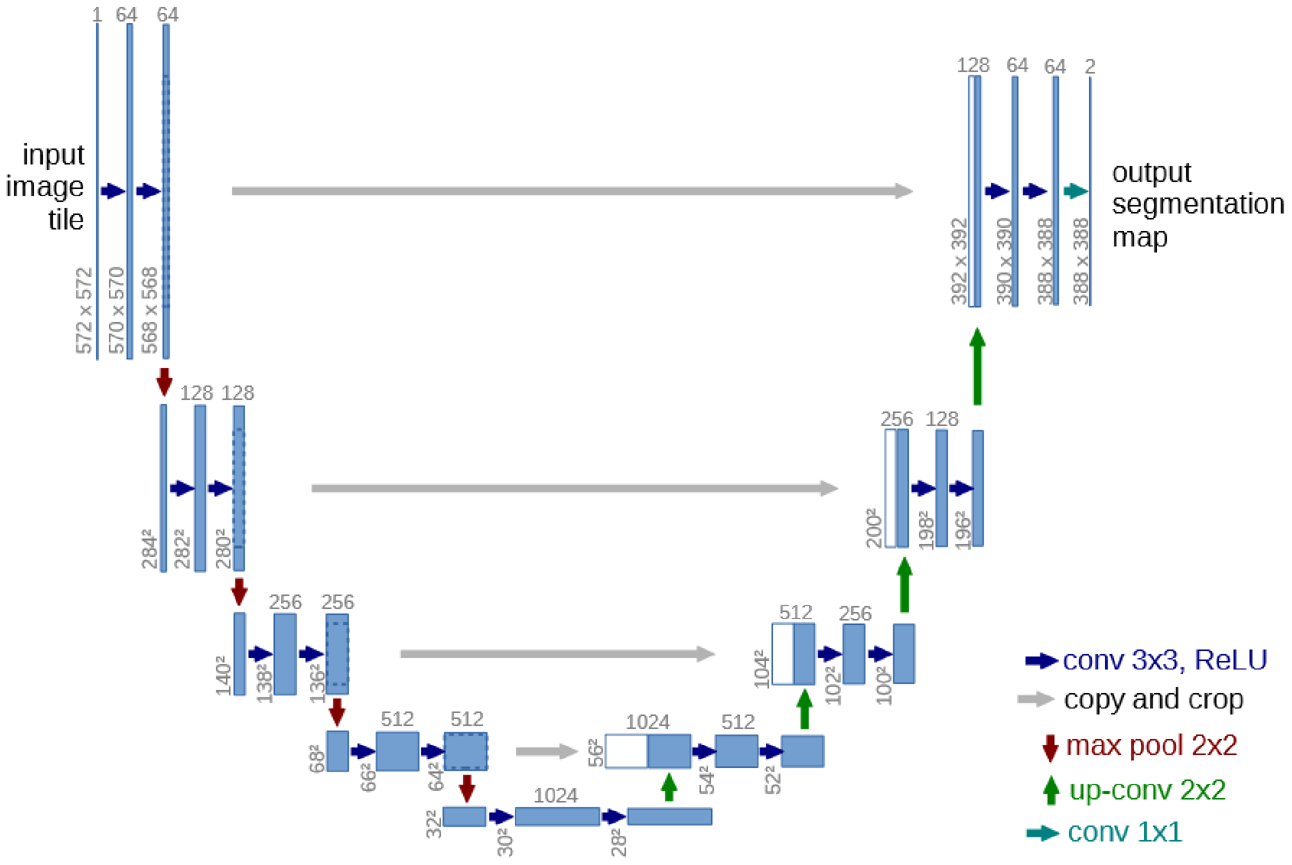
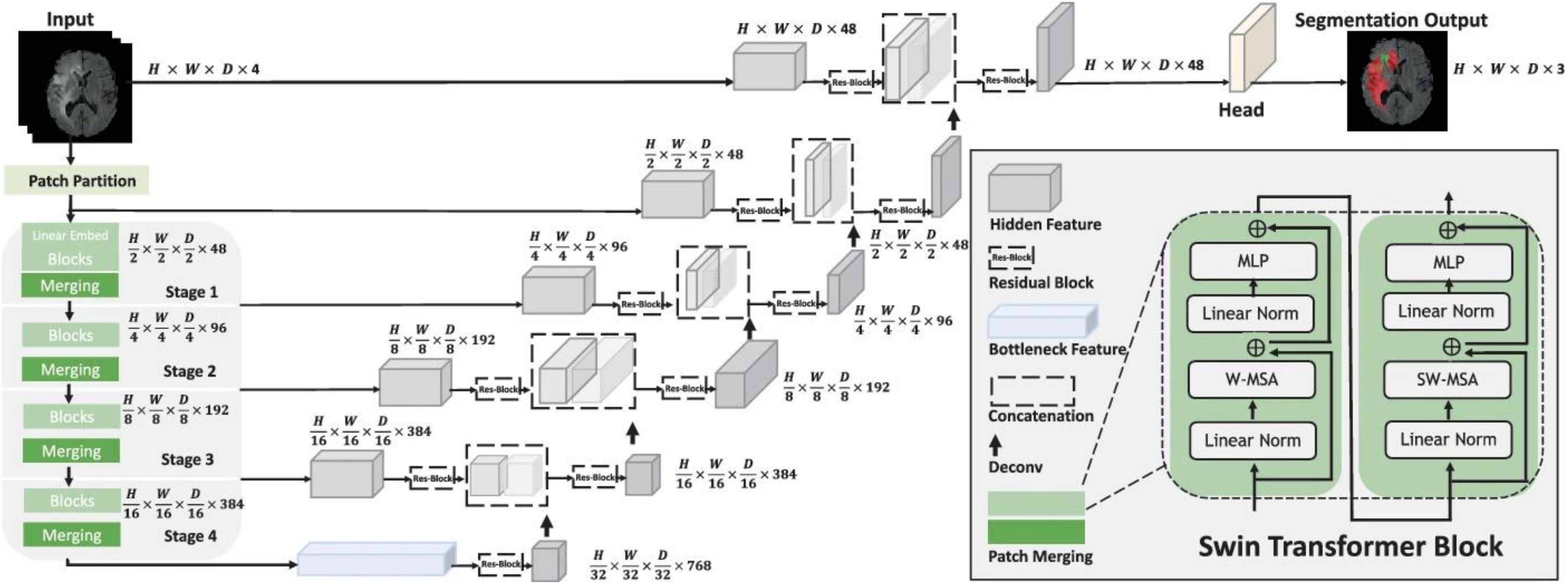


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

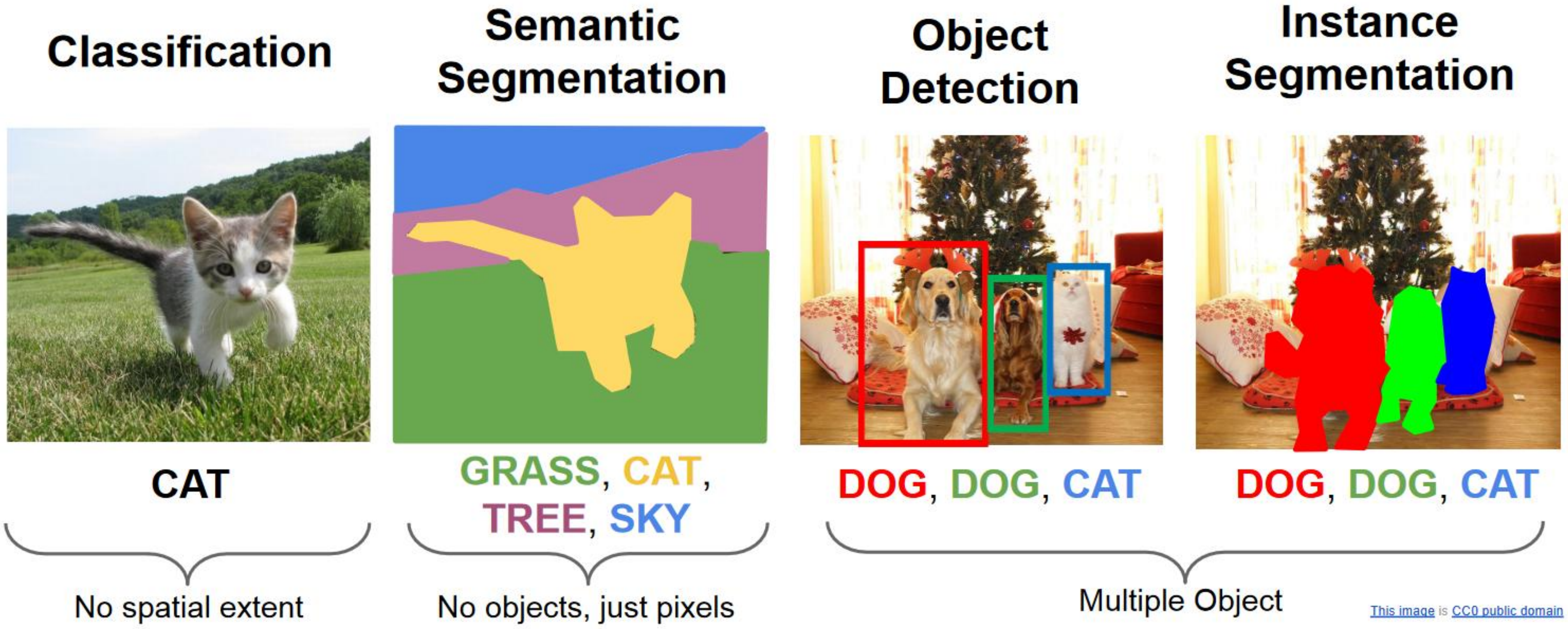
https://arxiv.org/pdf/1505.04597.pdf

Swin UNETR



https://link.springer.com/chapter/10.1007/978-3-031-08999-2_22

Instance segmentation



Source: cs231n slides

Mask R-CNN: An architecture for instance segmentation

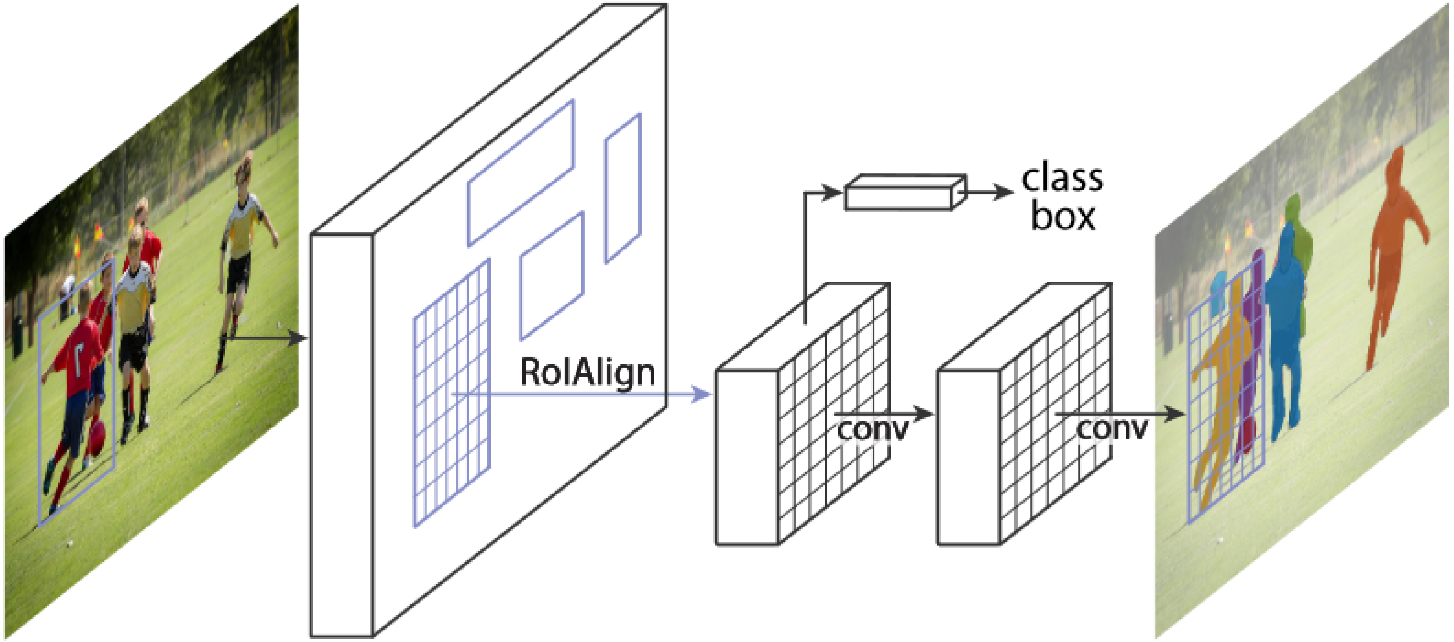


Figure 1. The **Mask R-CNN** framework for instance segmentation.

Extends Faster R-CNN by adding a branch for predicting segmentation masks on each Region of Interest (RoI), in parallel with the existing branch for classification and bounding box regression (Figure 1).
https://arxiv.org/pdf/1703.06870.pdf

Mask R-CNN example results

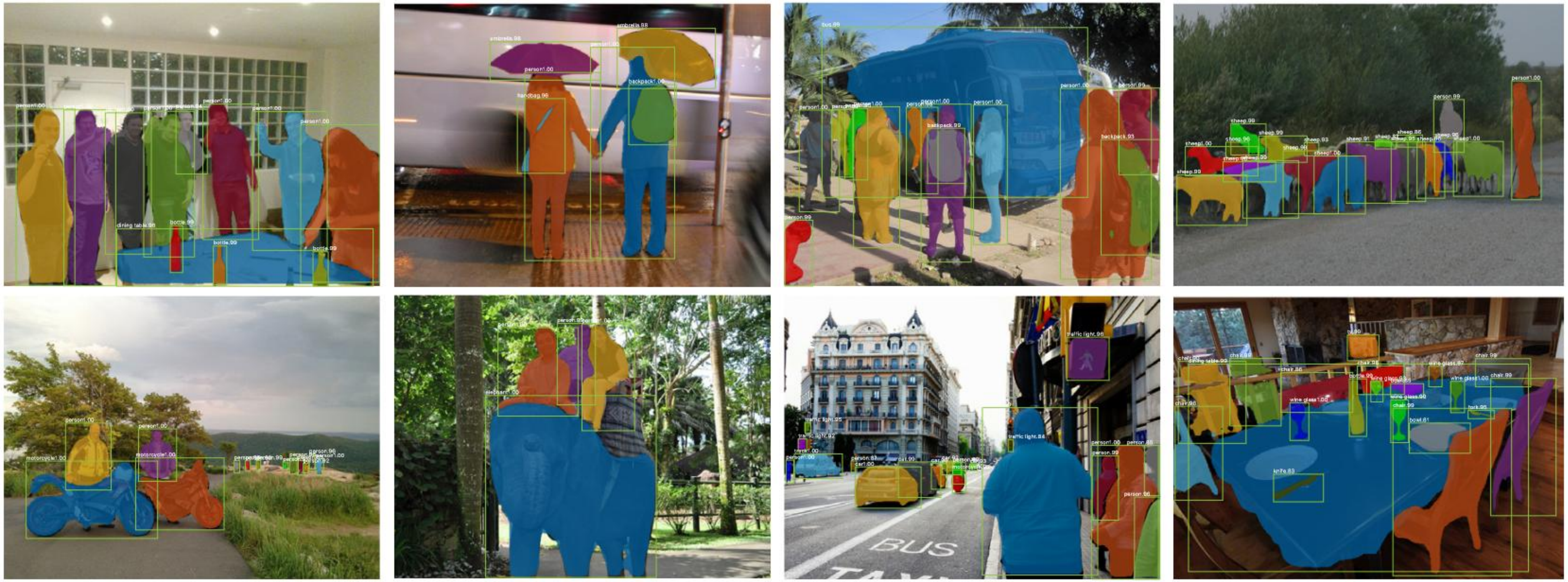


Figure 2. **Mask R-CNN** results on the COCO test set. These results are based on ResNet-101 [19], achieving a *mask* AP of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

https://arxiv.org/pdf/1703.06870.pdf

Implementation: https://github.com/facebookresearch/detectron2

SAM 2: State-of-the-art user interactive segmentation

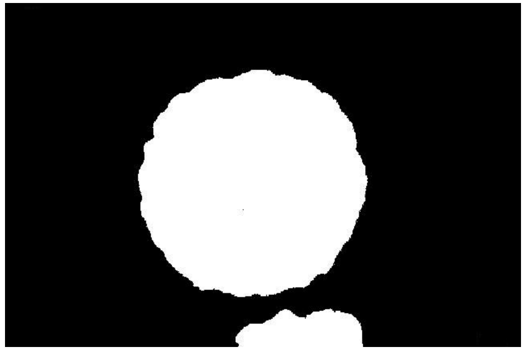
- https://ai.meta.com/sam2/

Semantic segmentation with PyTorch

We will do the simplest type of segmentation: foreground and background segmentation

Flower image

Semantic Segmentation



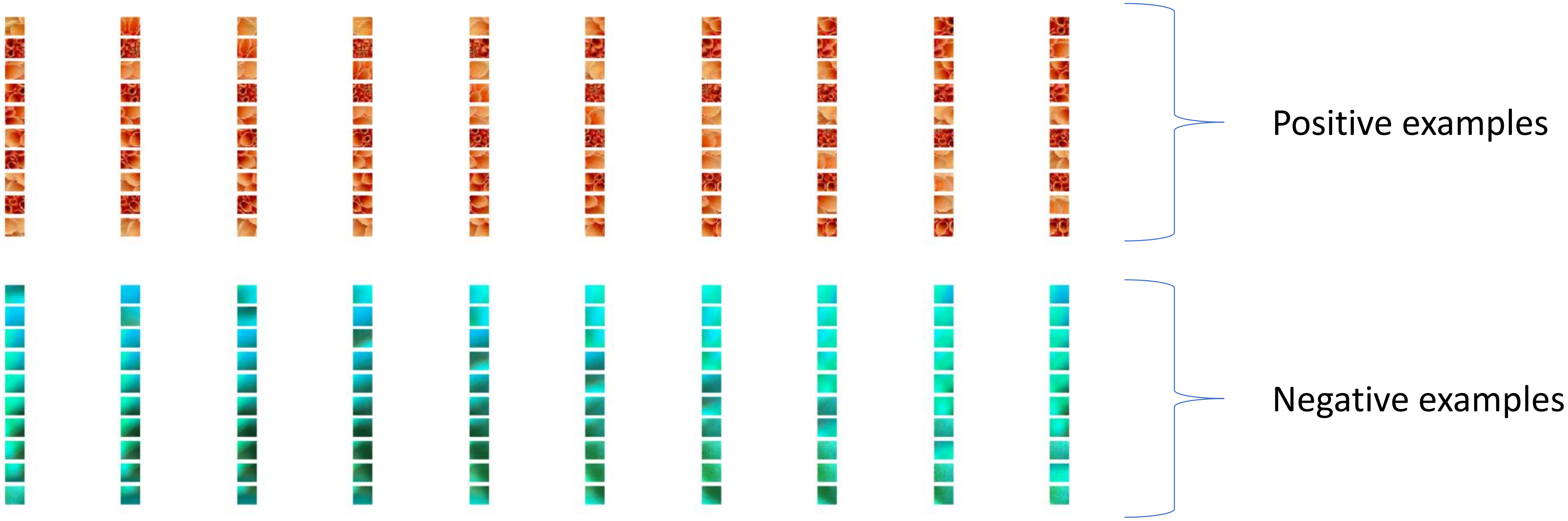
Binary foreground-background segmentation

Work with PyTorch for semantic segmentation

- We will look at two notebooks semantic segmentation
- For simplicity, we will do 2-class segmentation: foreground and background segmentation for a “flower.jpg” image
- Our first attempt will be to train a convnet on classification task and then use it for segmentation
    - In this process, we will learn one important feature: using a convolution in place of a fully connected layer
    - The advantage of using a fully convolutional layer is that the network can accept an image of any size as an input
- One limitation of applying a classification convent for segmentation is that the output would be smaller in size than the input
- Our next attempt will overcome this low-resolution issue by introducing an important type of convolution operation called “transposed convolution” (aka “upsampling”)
    - We will also learn about an important aspect about *labels* in training a “segmentation net” that is different from training a “classification net”

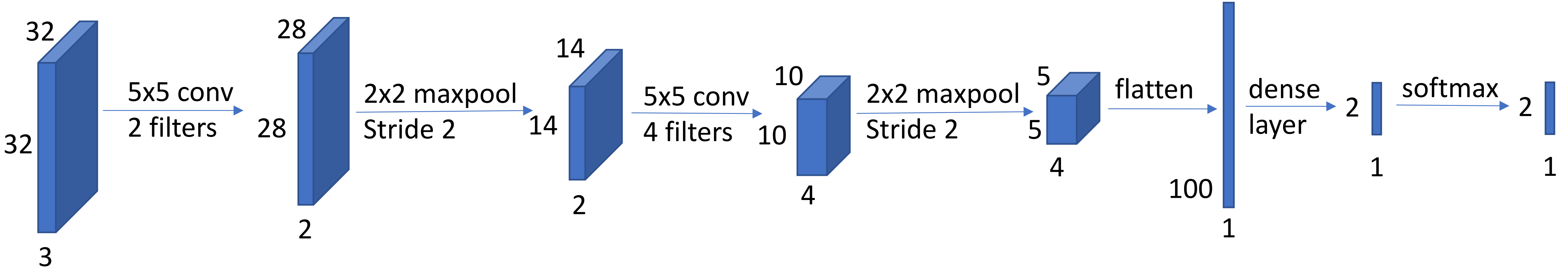
Segmentation as image classification

Training dataset: 32-by-32-by-3 RGB images with positive or negative labels



Segmentation as image classification…

Train a convnet on the training set

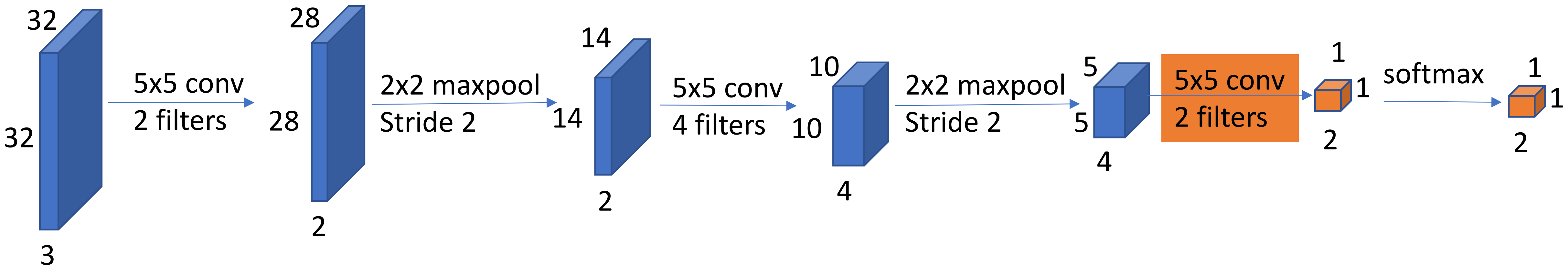


This classification net can only take a fixed size input image.

Segmentation_by_classification.ipynb

Segmentation as image classification…

Let’s convert the dense layer to a convolution layer:



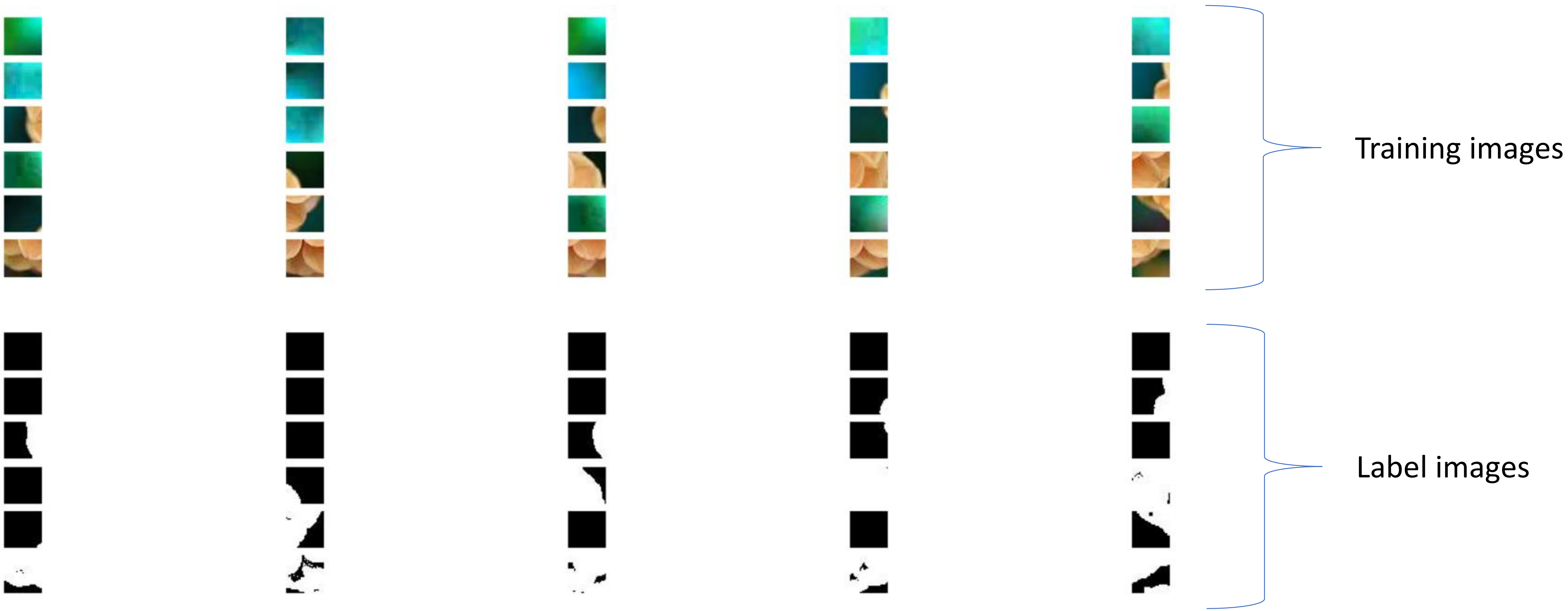
This classification net can take an input of any width and height.

But, output image will have approximately 4x lower resolution.

Segmentation using convolution transpose

- Convolution transpose is a type of convolution
  - See: https://nrupatunga.github.io/2016/05/14/convolution-arithmetic-in-deep-learning-part-1/
  - https://nrupatunga.github.io/convolution-2/
- Convolution transpose is more commonly known as “upsampling” layer
- Let’s call this net “Segmentation Net”

Training set for segmentation net

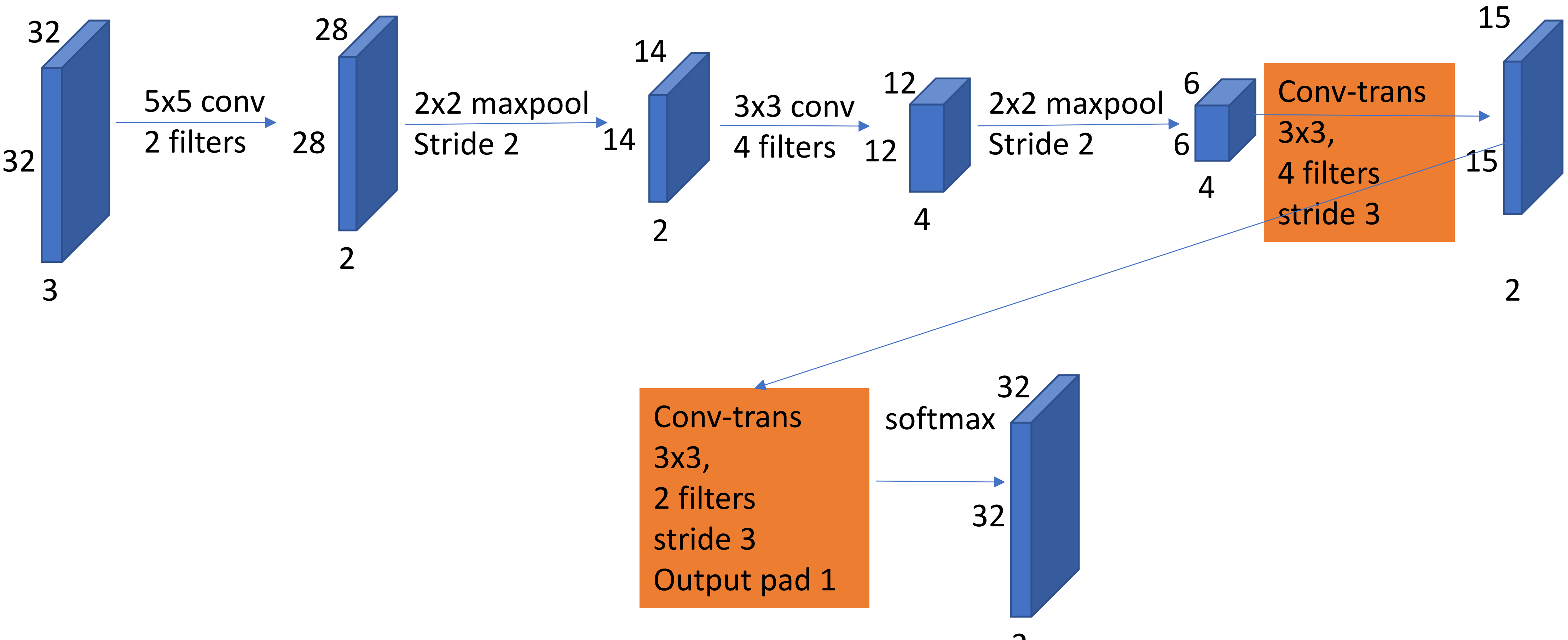


A training image has a corresponding label image of same height and width.

Segmentation net architecture

This network can take input of any height and width.

Output image does not lose resolution!



Convolution transpose layer

Roughly speaking, the size of the output of conv-trans layer would be determined by the following relationship:

output = conv(input, kernel, stride)

input = conv_trans(output, kernel, stride)

Skip connection – how to add one

Design this architecture and train; then segment the flower image with this architecture.